# using *make* to analyze data.

6/09/2016

P. Zumbo

# a brief history of make

- first introduced by Stuart Feldman in 1977 at Bell Labs
- traditionally been used to build executable programs and libraries from source code
  - however, make is **not** limited to building binaries and libraries

# what is make?

*make* is a program that reads a makefile and that builds one or more files from zero or more other files that they depend on.

# how does make do what it does?

make parses the makefile, builds a dependency tree (by determining the relationships between the inputs and outputs), and then traverses each branch of the tree, executing commands along the way.

# what is a makefile?

a *makefile* is a text file which contains *rules* for how to create a set of target files.

# what is a rule?

a rule tells *make* which series of commands to execute and what files must exist beforehand in order to create a set of targets from some input.

# the general form of a rule is:

```
target … : dependency …
   command
   …
   …
```

let’s see this in action….

# why use make? the limits of a script:

1. linear execution
   - make -j
2. truncated files
   - .DELETE_ON_ERROR:
3. unable to resume
   - make
4. poor audit trail
   - make -n > make.log